

What is claimed is:

1. An apparatus for collecting a profile of a subroutine in a program, the apparatus collecting the profile of said subroutine by using an interrupt generated when a branch instruction is executed during execution of said program.
2. An apparatus according to claim 1, which collects said profile for each executor of a subroutine.
3. An apparatus according to claim 2, which individually collects profiles of a plurality of subroutines executed by a specific executor.
4. An apparatus according to claim 3, which individually collects a first profile of a subroutine called by a main routine, and a second profile of this subroutine called by another subroutine.
5. An apparatus according to claim 4, which stores said second profile and calling relationship information relating to said second profile, said calling relationship information indicating a relationship between said other subroutine and said subroutine.
6. An apparatus according to claim 1, wherein said profile of said subroutine comprising at least one from a cumulative value of execution time, times of calling, time of final calling, and an overhead.
7. An apparatus according to claim 1 comprising:

a storage unit storing a profile of a subroutine; an analyzing section judging whether or not an executed branch instruction is an instruction relating to the execution of said subroutine when said interrupt is generated; and

a collecting section, when said analyzing section identifies that said executed branch instruction is the instruction relating to the execution of said subroutine, for obtaining a profile of said subroutine and stores said obtained profile in said storage unit.

8. An apparatus according to claim 7, wherein a plurality of storage unit respectively corresponding to a plurality of executors of said subroutine are prepared; and

    said collecting section specifies said executor of said subroutine and stores said profile of said subroutine corresponding to said specified executor in said storage unit.

9. An apparatus according to claim 8, wherein said collecting section individually stores profiles of a plurality of subroutines corresponding to a specified executor in said storage unit.

10. An apparatus according to claim 9, wherein said collecting section individually stores a first profile of a subroutine called by a main routine, and a second profile of

the subroutine called by another subroutine, in said storage unit.

11. An apparatus according to claim 10, wherein said collecting section stores said second profile and calling relationship information relating to said second profile, said calling relationship information indicating a relationship between said other subroutine and said called subroutine, in said storage unit.

12. An apparatus according to claim 1, further comprising:

a storage unit storing a profile;  
an analyzing section, when said interrupt generates, obtaining a branch source address and a branch destination address from a source of said interrupt, and identifying a type of said brunch instruction by obtaining a instruction code from said brunch source address and decoding said instruction code; and

a collecting section obtaining said brunch source address, said brunch destination address, and a identified result from said analyzing section when the identified instruction is a calling instruction or a return instruction of said subroutine; when said identified result is said calling instruction, storing said brunch destination address as a subroutine address corresponding to said calling instruction and a calling time of said subroutine

corresponding to said calling instruction in said storage unit; and when said identified result is said return instruction, obtaining a return time of said subroutine corresponding to said return instruction, calculating a execution time of said subroutine based on said obtained return time and said calling time, and storing a cumulative value of said execution time as said profile in correspondence with said branch destination address in said storage unit.

13. An apparatus according to claim 12, wherein said collecting section stores times of calling of said subroutine corresponding to said brunch destination address as said profile in said storage unit.

14. An apparatus according to claim 12, wherein said collecting section obtains an overhead of said subroutine as said profile and stores said overhead in said storage unit.

15. An apparatus according to claim 13, wherein said collecting section, when said identified result is said calling instruction, stores an identifier of an executor of said subroutine corresponding to said calling instruction and said brunch destination address in said storage unit.

16. An apparatus according to claim 13, wherein said collecting section, when said identified result is said calling instruction and said brunch source address and said branch destination address are addresses of said subroutines,

stores said branch source address and branch destination address as calling relationship information indicating a calling source subroutine and a calling destination subroutine in said storage unit, and stores at least one of the cumulative execution time and the times of calling in said calling destination subroutine in the call source subroutine, as said profile corresponding to said calling relationship information, in said storage unit.

17. An apparatus according to claim 7, further comprising a setting section setting an execution environment of a source of said interrupt so as to generate said interrupt when said branch instruction is executed during the execution of said program.

18. A computer readable medium stores a program in order that a computer executes a process for collecting a profile of a subroutine included another program as an analyzing target, said program comprising steps of:

identifying, when an interrupt is generated by an execution of a brunch instruction during the execution said other program, whether or not said executed brunch instruction is an instruction relating to the execution of said subroutine;

obtaining, when said brunch instruction is the instruction relating to the execution of said subroutine, said profile of said subroutine; and

storing said profile in a storage unit.

19. A computer readable medium according to claim 19, wherein said program further comprises steps of:

specifying an executor of said subroutine; and

storing said profile in a storage unit corresponding to said specified executor within a plurality of storage unit provided with each executor.

20. A computer readable medium according to claim 19, wherein said program further comprises step of storing, when a specified executor executes a plurality of subroutines, profiles of said plurality of subroutines in said storage unit corresponding to said specified executor.

21. A computer readable medium according to claim 20, wherein said program further comprises step of storing a first profile of said subroutine called by a main routine and a second profile of said subroutine called by another subroutine in said corresponding storage unit, regarding to each of said subroutines.

22. A computer readable medium according to claim 21, wherein said program further comprises step of storing said second profile and calling relationship information relating to said second profile, said calling relationship information indicating a relationship between said other subroutine and said subroutine.

23. A computer readable medium according to claim 18, wherein said program further comprises steps of:

obtaining a branch source address and a branch destination address from a source of said interrupt when said interrupt generates;

identifying a type of said branch instruction by obtaining a instruction code from said branch source address and decoding said instruction code;

storing said branch destination address as a subroutine address corresponding to said calling instruction and a calling time of said subroutine corresponding to said calling instruction in said storage unit when said identified result is said calling instruction; and,

when said identified result is said return instruction, obtaining a return time of said subroutine corresponding to said return instruction, calculating a execution time of said subroutine based on said obtained return time and said calling time, and storing a cumulative value of said execution time as said profile in correspondence with said branch destination address in said storage unit.

24. A computer readable medium according to claim 23, wherein said program further comprises step of storing times of calling of said subroutine corresponding to said branch destination address as said profile in said storage unit.

25. A computer readable medium according to claim 23, wherein said program further comprises step of storing an overhead of said subroutine as said profile in said storage unit.

26. A computer readable medium according to claim 23, wherein said program further comprises step of storing, when said identified result is said calling instruction, an identifier of an executor of said subroutine corresponding to said calling instruction and said brunch destination address in said storage unit.

27. A computer readable medium according to claim 23, wherein said program further comprises steps of, when said identified result is said calling instruction and said brunch source address and said branch destination address are addresses of said subroutines, storing said branch source address and branch destination address as calling relationship information indicating a calling source subroutine, and a calling destination subroutine in said storage unit; and storing at least one of the cumulative execution time and the times of calling in said calling destination subroutine in the call source subroutine, as said profile corresponding to said calling relationship information, in said storage unit.

28. A computer readable medium according to claim 18, wherein said program further comprises step of setting an

execution environment of a source of said interrupt so as to generate said interrupt when said branch instruction is executed during the execution of said program.

29. A method for collecting a profile of a subroutine in a program, comprising steps of:

identifying, when an interrupt is generated by an execution of a branch instruction during the execution of the program, whether or not the executed branch instruction is an instruction relating to the execution of the subroutine;

obtaining, when the branch instruction is the instruction relating to the execution of the subroutine, the profile of the subroutine; and

storing the profile in a storage unit.

30. A method according to claim 29, further comprising steps of:

specifying an executor of the subroutine; and

storing the profile in a storage unit corresponding to the specified executor within a plurality of storage unit provided with each executor.

31. A method according to claim 30, further comprising step of storing, when a specified executor executes a plurality of subroutines, profiles of the plurality of subroutines in the storage unit corresponding to the specified executor.

32. A method according to claim 31, further comprising step of storing a first profile of the subroutine called out by a main routine and a second profile of the subroutine called out by another subroutine in the corresponding storage unit, regarding to each of the subroutines.

33. A method according to claim 32, further comprising step of storing the second profile and calling relationship information relating to the second profile, the calling relationship information indicating a relationship between the other subroutine and the subroutine.

34. A method according to claim 29, further comprising steps of:

obtaining a branch source address and a branch destination address from a source of the interrupt when the interrupt generates;

identifying a type of the branch instruction by obtaining a instruction code from the branch source address and decoding the instruction code;

storing the branch destination address as a subroutine address corresponding to the calling instruction and a calling time of the subroutine corresponding to the calling instruction in the storage unit when the identified result is the calling instruction; and

when the identified result is the return instruction, obtaining a return time of the subroutine corresponding to

the return instruction, calculating a execution time of the subroutine based on the obtained return time and the calling time, and storing a cumulative value of the execution time as the profile in correspondence with the branch destination address in the storage unit.

35. A method according to claim 34, further comprising step of storing times of calling of the subroutine corresponding to the brunch destination address as the profile in said storage unit.

36. A method according to claim 34, further comprising step of storing an overhead of the subroutine as the profile in the storage unit.

37. A method according to claim 34, further comprising step of storing, when the identified result is the calling instruction, an identifier of an executor of the subroutine corresponding to the calling instruction and the brunch destination address in the storage unit.

38. A method according to claim 34, further comprising steps of, when the identified result is the calling instruction and the brunch source address and the branch destination address are addresses of the subroutines, storing the branch source address and the branch destination address as calling relationship information indicating a calling source subroutine, and a calling destination subroutine in said storage unit; and storing at least one of the cumulative

execution time and the times of calling in the calling destination subroutine in the call source subroutine, as the profile corresponding to the calling relationship information, in the storage unit.

39. A method according to claim 29, further comprising step of setting an execution environment of a source of the interrupt so as to generate the interrupt when the branch instruction is executed during the execution of the program.